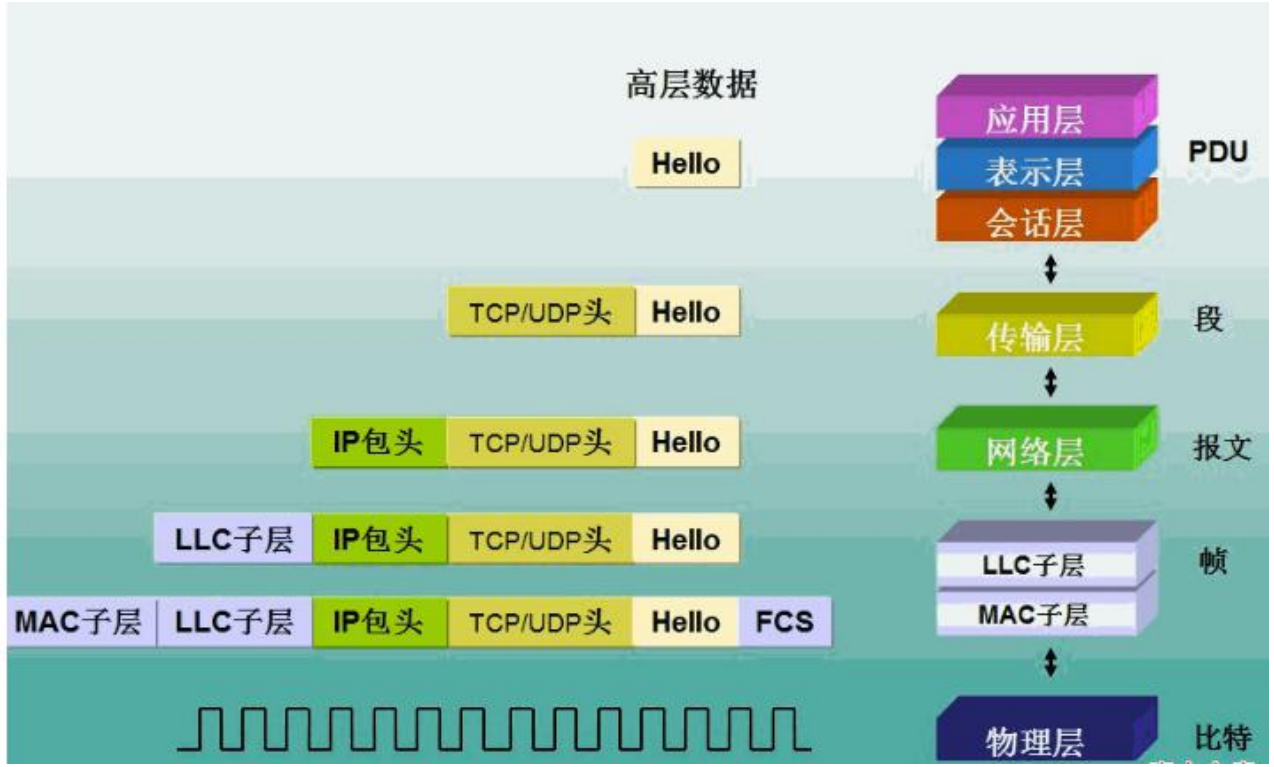
数据流程


高层数据
Hello
TCP/UDP头
Hello
IP包头
TCP/UDP头
Hello
LLC子层
IP包头
TCP/UDP头
Hello
MAC子层
LLC子层
IP包头
TCP/UDP头
Hello
FCS
应用层
表示层
会话层
PDU
传输层
段
网络层
报文
LLC子层
MAC子层
帧
物理层
比特

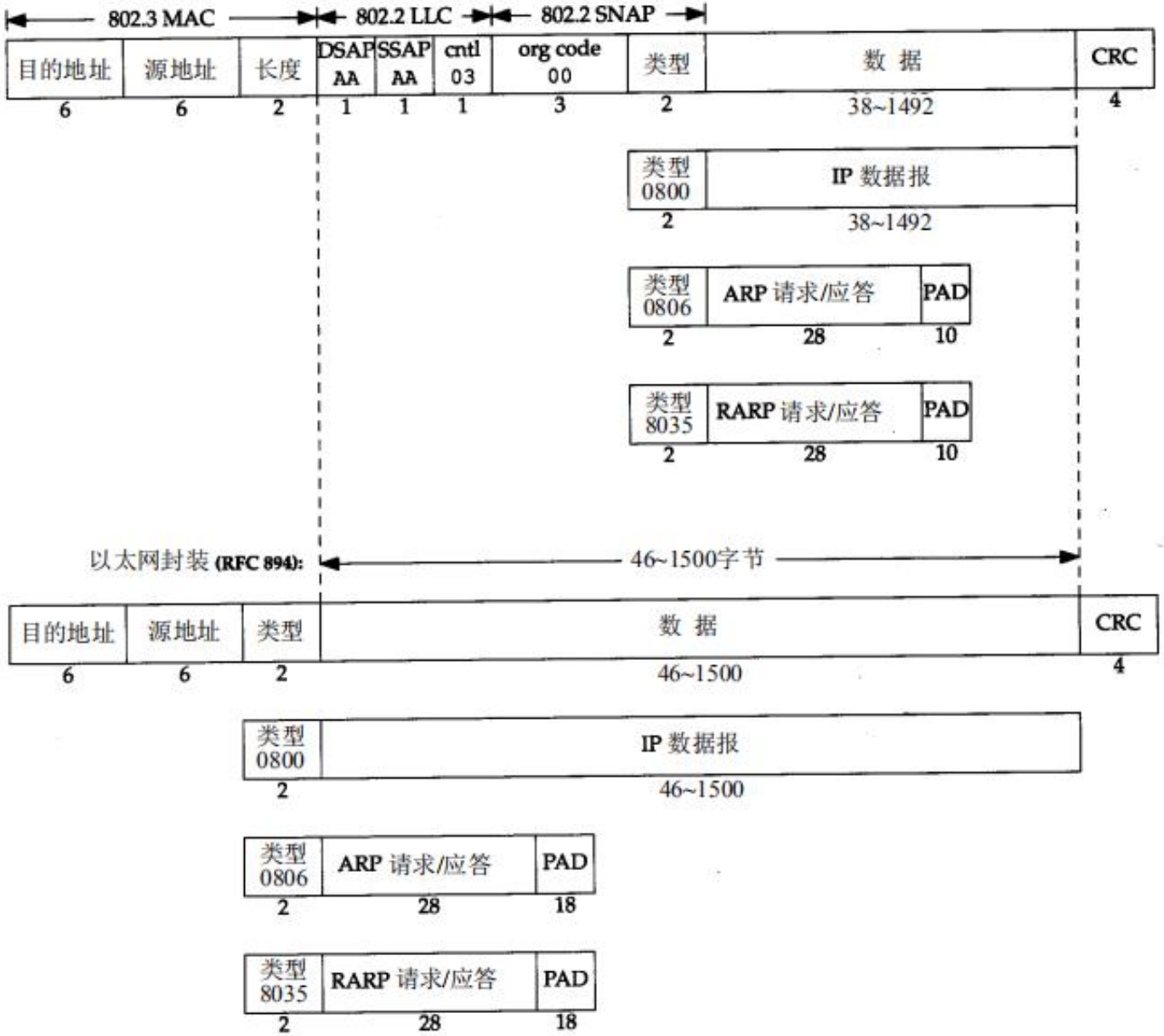
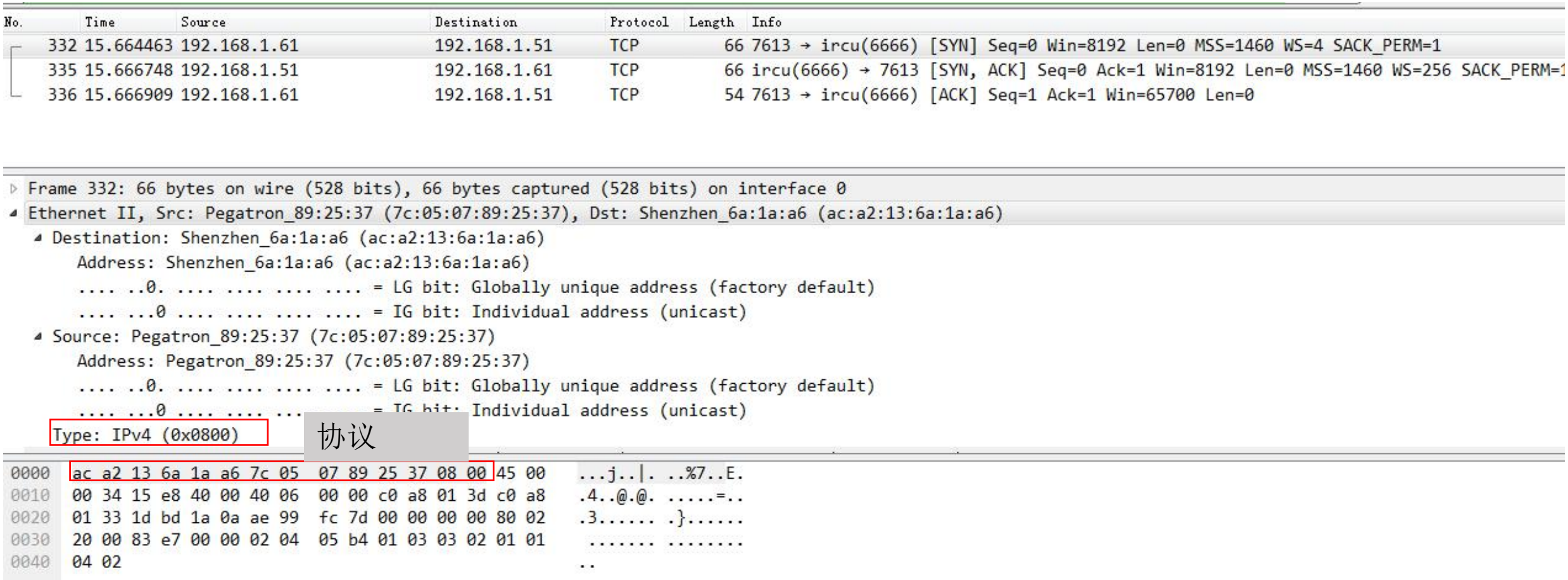


图2-1 IEEE 802.2/802.3（RFC 1042）和以太网的封装格式（RFC 894）

注：CRC字段用于帧内后续字节差错的循环冗余码检验（检验和）（它也被成为FCS或者帧检验序列）


No. Time Source Destination Protocol Length Info
332 15.664463 192.168.1.61 192.168.1.51 TCP 66 7613 → ircu(6666) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
335 15.666748 192.168.1.51 192.168.1.61 TCP 66 ircu(6666) → 7613 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
336 15.666909 192.168.1.61 192.168.1.51 TCP 54 7613 → ircu(6666) [ACK] Seq=1 Ack=1 Win=65700 Len=0
Frame 332: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Pegatron_89:25:37 (7c:05:07:89:25:37), Dst: Shenzhen_6a:1a:a6 (ac:a2:13:6a:1a:a6)
Destination: Shenzhen_6a:1a:a6 (ac:a2:13:6a:1a:a6)
Address: Shenzhen_6a:1a:a6 (ac:a2:13:6a:1a:a6)
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Source: Pegatron_89:25:37 (7c:05:07:89:25:37)
Address: Pegatron_89:25:37 (7c:05:07:89:25:37)
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
协议
0000 ac a2 13 6a 1a a6 7c 05 07 89 25 37 08 00 45 00 ...j..|. ..%7..E.
0010 00 34 15 e8 40 00 40 06 00 00 c0 a8 01 3d c0 a8 .4..@.@. .....=..
0020 01 33 1d bd 1a 0a ae 99 fc 7d 00 00 00 00 80 02 .3..... .}......
0030 20 00 83 e7 00 00 02 04 05 b4 01 03 03 02 01 01 ....
0040 04 02 ..

3.2 IP首部

IP数据报的格式如图3-1所示。普通的IP首部长为20个字节，除非含有选项字段。

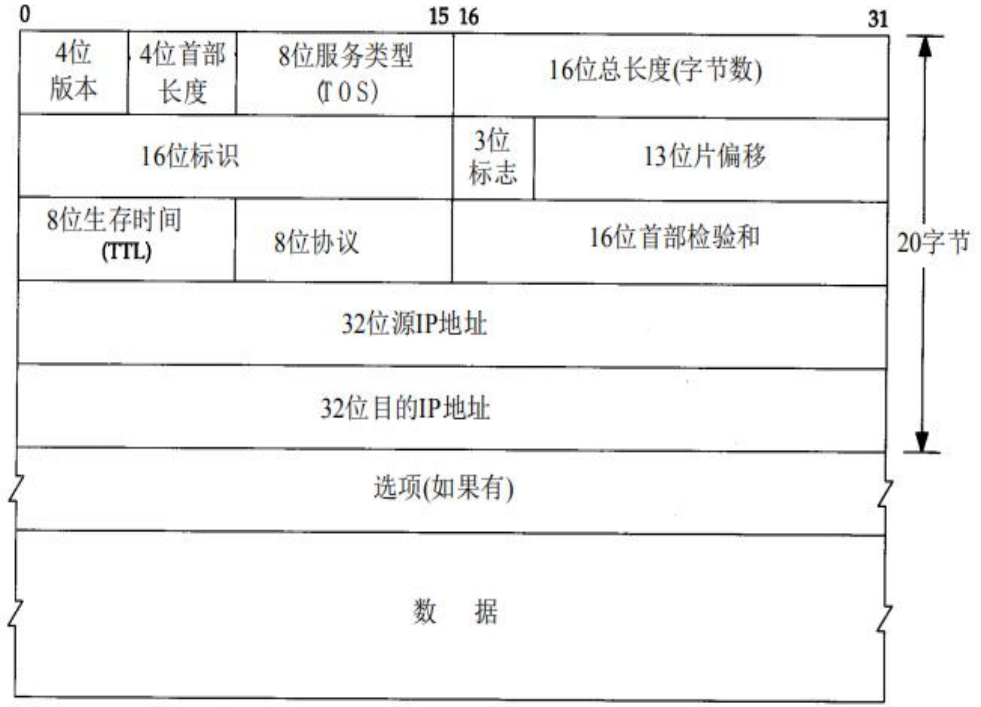
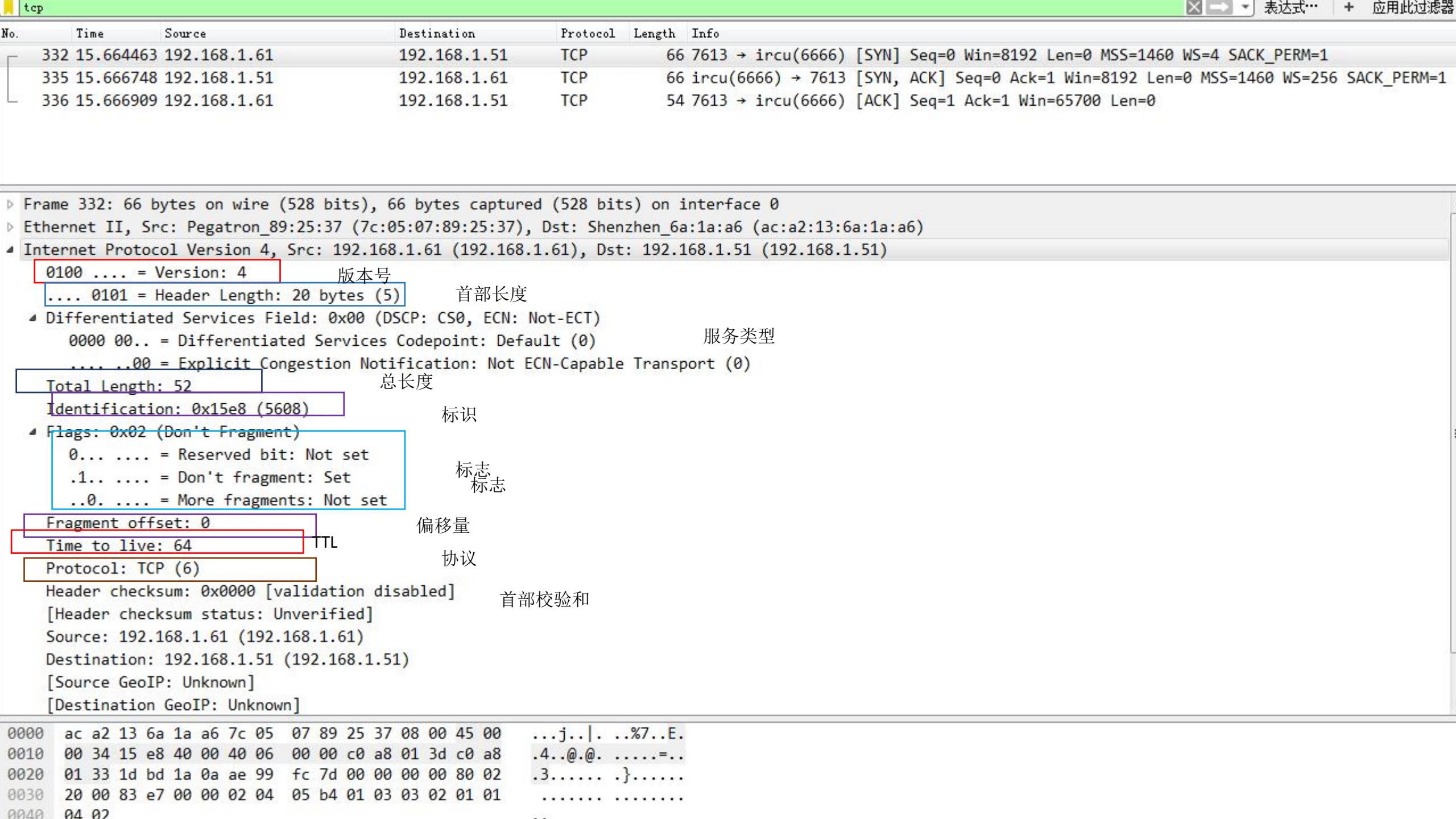


图3-1 IP数据报格式及首部中的各字段

应用程序	最小时延	最大吞吐量	最高可靠性	最小费用	16进制值
Telnet/Rlogin	1	0	0	0	0x10
FTP					
控制	1	0	0	0	0x10
数据	0	1	0	0	0x08
任意块数据	0	1	0	0	0x08
TFTP	1	0	0	0	0x10
SMTP					
命令阶段	1	0	0	0	0x10
数据阶段	0	1	0	0	0x08
DNS					
UDP查询	1	0	0	0	0x10
TCP查询	0	0	0	0	0x00
区域传输	0	1	0	0	0x08
ICMP					
差错	0	0	0	0	0x00
查询	0	0	0	0	0x00
任何IGP	0	0	1	0	0x04
SNMP	0	0	1	0	0x04
BOOTP	0	0	0	0	0x00
NNTP	0	0	0	1	0x02

图3-2 服务类型字段推荐值

- 首部长指的是首部占32bit字的长度，包括任何选项因是4bit字段，故首部最长为60个字节。
- 总长度字段：IP数据报的长度，以字节为单位。
- 据1和2可以知道IP数据包中数据内容的起始位置和长度。
- TTL:数据报可以经过的最多路由器数目，源主机设置，经过一个处理的路由器，其值减1


tcp
表达式…
应用此过滤器
No. Time Source Destination Protocol Length Info
332 15.664463 192.168.1.61 192.168.1.51 TCP 66 7613 → ircu(6666) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
335 15.666748 192.168.1.51 192.168.1.61 TCP 66 ircu(6666) → 7613 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
336 15.666909 192.168.1.61 192.168.1.51 TCP 54 7613 → ircu(6666) [ACK] Seq=1 Ack=1 Win=65700 Len=0
Frame 332: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Pegatron_89:25:37 (7c:05:07:89:25:37), Dst: Shenzhen_6a:1a:a6 (ac:a2:13:6a:1a:a6)
Internet Protocol Version 4, Src: 192.168.1.61 (192.168.1.61), Dst: 192.168.1.51 (192.168.1.51)
0100 .... = Version: 4
版本号
.... 0101 = Header Length: 20 bytes (5)
首部长度
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
服务类型
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 52
总长度
Identification: 0x15e8 (5608)
标识
Flags: 0x02 (Don't Fragment)
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment: Set
标志
标志
..0. .... = More fragments: Not set
Fragment offset: 0
偏移量
Time to live: 64
TTL
Protocol: TCP (6)
协议
Header checksum: 0x0000 [validation disabled]
首部校验和
[Header checksum status: Unverified]
Source: 192.168.1.61 (192.168.1.61)
Destination: 192.168.1.51 (192.168.1.51)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
0000 ac a2 13 6a 1a a6 7c 05 07 89 25 37 08 00 45 00 ...j..|. ..%7..E.
0010 00 34 15 e8 40 00 40 06 00 00 c0 a8 01 3d c0 a8 .4..@.@. .....=..
0020 01 33 1d bd 1a 0a ae 99 fc 7d 00 00 00 00 80 02 .3...... .}......
0030 20 00 83 e7 00 00 02 04 05 b4 01 03 03 02 01 01 ....
0040 04 02

17.3 TCP的首部

TCP数据被封装在一个IP数据报中，如图17-1所示。

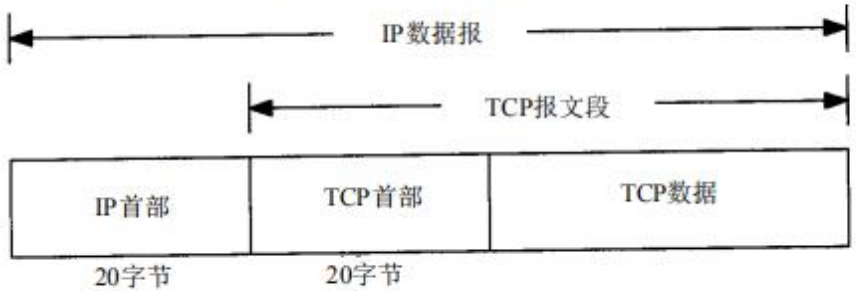


图17-1 TCP数据在IP数据报中的封装

图17-2显示TCP首部的数据格式。如果不计任选字段，它通常是20个字节。

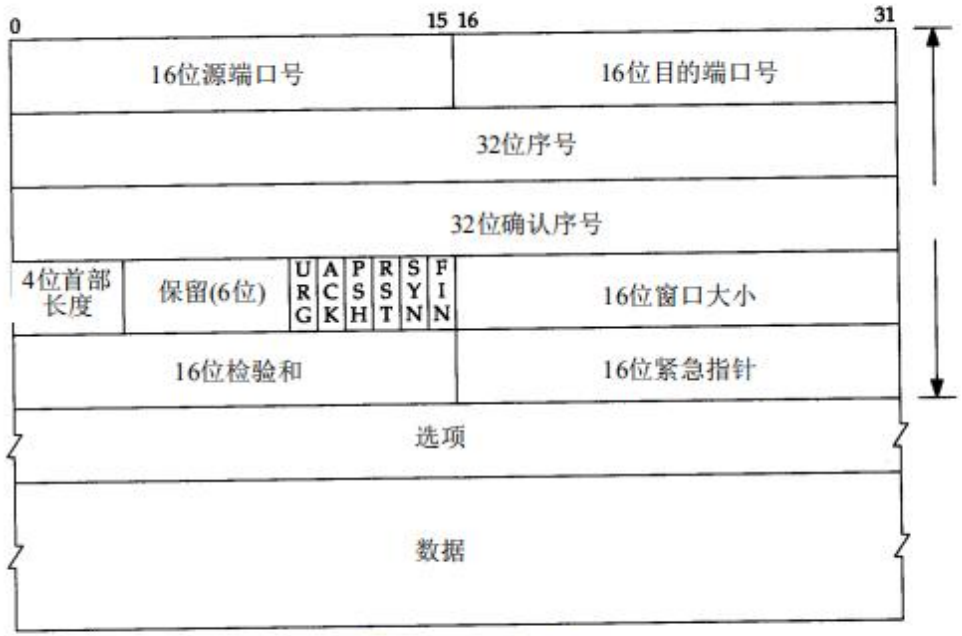


图17-2 TCP包首部

标志(6bit)

在TCP首部中有6个标志比特，他们中的多个可同时被置为1

- URG(Urgent Pointer Field Significant):紧急指针标志，用来保证TCP连接不被中断，并且督促中间设备尽快处理这些数据
- ACK(Acknowledgement Field Signigicant):确认号字段，该字段为1时表示应答字段有效，即TCP应答号将包含在TCP报文中。
- PSH(Push Function): 推送功能，所谓推送功能指的是接收端在接收到数据后立即推送给应用程序，而不是在缓冲区中排队。
- RST(Reset the connection): 重置连接，不过一搬表示断开一个连接，

- SYN(Synchronize sequence numbers):同步序列号，用来发起一个连接请求
- FIN(No more data from sender):表示发送端发送任务已经完成（既断开连接）

```
332 15.664463 192.168.1.61   192.168.1.51   TCP   66 7613 → ircu(6666) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
335 15.666748 192.168.1.51   192.168.1.61   TCP   66 ircu(6666) → 7613 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
336 15.666909 192.168.1.61   192.168.1.51   TCP   54 7613 → ircu(6666) [ACK] Seq=1 Ack=1 Win=65700 Len=0
```

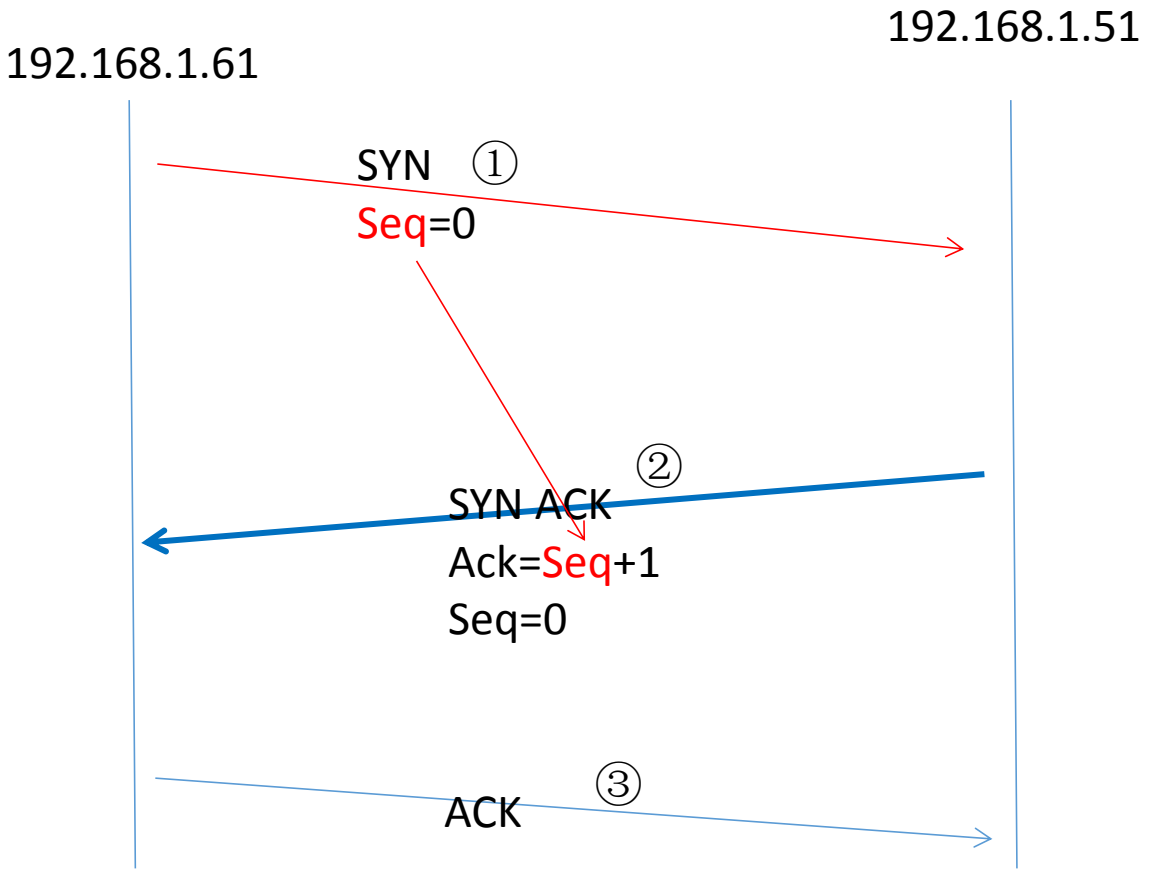
```
▷ Internet Protocol Version 4, Src: 192.168.1.61 (192.168.1.61), Dst: 192.168.1.51 (192.168.1.51)
◢ Transmission Control Protocol, Src Port: 7613 (7613), Dst Port: ircu (6666), Seq: 0, Len: 0
    Source Port: 7613 (7613)                          源端口
    Destination Port: ircu (6666)                     目标端口
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)  序号
    Acknowledgment number: 0                          确认序号
    1000 .... = Header Length: 32 bytes (8)           首部长度
  ◢ Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set                标志
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ▷ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····S·]
    Window size value: 8192                           窗口大小
    [Calculated window size: 8192]
    Checksum: 0x83e7 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ◢ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
    ▷ TCP Option - Maximum segment size: 1460 bytes
    ▷ TCP Option - No-Operation (NOP)
    ▷ TCP Option - Window scale: 2 (multiply by 4)
```

```
0000  ac a2 13 6a 1a a6 7c 05  07 89 25 37 08 00 45 00   ...j..|. ..%7..E.
0010  00 34 15 e8 40 00 40 06  00 00 c0 a8 01 3d c0 a8   .4..@.@. .....=..
```

tcp三次握手

tcp 表达式… + 应用此过滤器

No.	Time	Source	Destination	Protocol	Length	Info
332	15.664463	192.168.1.61	192.168.1.51	TCP	66	7613 → ircu(6666) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
335	15.666748	192.168.1.51	192.168.1.61	TCP	66	ircu(6666) → 7613 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
336	15.666909	192.168.1.61	192.168.1.51	TCP	54	7613 → ircu(6666) [ACK] Seq=1 Ack=1 Win=65700 Len=0



18.2.4 连接终止协议

建立一个连接需要三次握手，而终止一个连接要经过 4次握手。这由TCP的半关闭（half-close）造成的。既然一个TCP连接是全双工（即数据在两个方向上能同时传递），因此每个方向必须单独地进行关闭。这原则就是当一方完成它的数据发送任务后就能发送一个 FIN来终止这个方向连接。当一端收到一个 FIN，它必须通知应用层另一端几经终止了那个方向的数据传送。发送FIN通常是应用层进行关闭的结果。

收到一个FIN只意味着在这一方向上没有数据流动。一个 TCP连接在收到一个FIN后仍能发送数据。而这对利用半关闭的应用来说是可能的，尽管在实际应用中只有很少的 TCP应用程序这样做。正常关闭过程如图 18-3所示。我们将在18.5节中详细介绍半关闭。

首先进行关闭的一方（即发送第一个 FIN）将执行主动关闭，而另一方（收到这个 FIN）执行被动关闭。通常一方完成主动关闭而另一方完成被动关闭，但我们将在 18.9节看到双方如何都执行主动关闭。

图18-3中的报文段4发起终止连接，它由Telnet客户端关闭连接时发出。这在我们键入 quit命令后发生。它将导致TCP客户端发送一个FIN，用来关闭从客户到服务器的数据传送。

当服务器收到这个 FIN，它发回一个ACK，确认序号为收到的序号加 1（报文段5）。和SYN一样，一个FIN将占用一个序号。同时 TCP服务器还向应用程序（即丢弃服务器）传送一个文件结束符。接着这个服务器程序就关闭它的连接，导致它的 TCP端发送一个FIN（报文段6），客户必须发回一个确认，并将确认序号设置为收到序号加1（报文段7）。

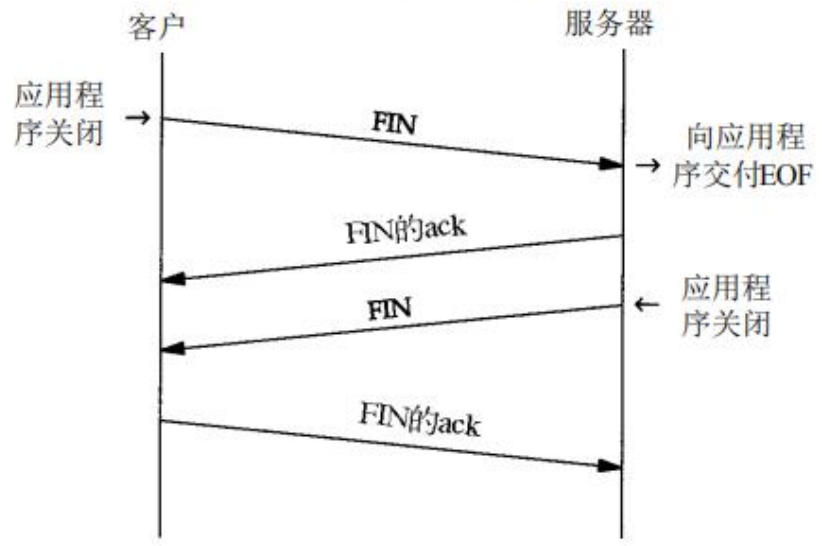


图18-4显示了终止一个连接的典型握手